Problem A

National Science Olympiad

In the recent National Science Olympiad, a new tie-breaker rule was implemented using time penalties measured in minutes.

The olympiad ran for two days and participated in by N contestants. The i-th participant, whose name is S_i, received A_i points with time penalty B_i on the first day, and received C_i points with time penalty D_i on the second day.

The contestants are ranked based on their total points from both days, with **higher points** being ranked higher. Contestants who receive the same total points will be ranked based on the sum of their time penalties from both days, with **lower penalties** being ranked higher. If there are still ties, they will be ranked based on their name, with the **lexicographically smaller name** being ranked higher.

Your task is to output the name of the contestants ranked from highest to lowest.

Input

The first line contains an integer N $(1 \leq N \leq 100)$. Each of the next N lines contains a string S_i $(1 \leq |S_i| \leq 10)$, containing uppercase English alphabets, followed by four integers A_i, B_i, C_i, and D_i $(0 \leq A_i, B_i, C_i, D_i \leq 300)$ representing their points and penalties from both days.

Output

Output N lines, each containing the name of the contestants, in order from the higher-ranked to the lower-ranked.

Sample Input 1

```
5
ANDI 200 120 150 130
BUDI 170 70 180 170
CUPU 0 300 0 300
DEWA 300 0 300 0
MALANG 0 300 0 300
```

Sample Output 1

```
DEWA
BUDI
ANDI
CUPU
MALANG
```

Explanation of Sample 1: Let us consider the ranks between ANDI and BUDI. Both ANDI and BUDI received the same total points, which is 350, but BUDI is ranked higher because BUDI's time penalty (which is 240) is lower than ANDI's (which is 250).

This page is intentionally left blank.

Problem B
Construct BFS Graph

You are currently researching a graph traversal algorithm called the Breadth First Search (BFS). Suppose there is a graph of N nodes, numbered from 1 to N, and an adjacency matrix A, for which node u can traverse to node v if $A_{u,v}$ is 1, otherwise it is 0. The following pseudocode will output the order the nodes that are visited in a BFS algorithm.

```
BFS(A[1..N][1..N]):
    let U be an empty array
    let Q be an empty queue

    append 1 to U
    push 1 to Q

    while Q is not empty:
        pop the front element of Q into u
        for v = 1 to N:
            if A[u][v] == 1 and v is not in U:
                append v to U
                push v to Q

    return U
```

Suppose now you have an integer N, M, and an array U of N integers. You wonder whether there exists a simple undirected graph with N nodes and M edges such that the output of the pseudocode above is the array U. Construct such graph if it exists.

A simple undirected graph with M edges has an adjacency matrix A that satisfies the following.

- $A_{u,u} = 0$ for all $1 \leq u \leq N$.
- Exactly M pairs (u, v) satisfies $1 \leq u < v \leq N$ and $A_{u,v} = 1$, meaning that there is an edge connecting node u and v.
- $A_{u,v} = A_{v,u}$ for all $1 \leq u < v \leq N$.

Input

The first line contains two integers N and M ($1 \leq N, M \leq 200\,000$). The second line contains N integers representing U, which is a permutation of $(1, 2, \ldots, N)$. You are guaranteed that the first element of U is always 1.

Output

If such a graph exists, output M lines, each containing two integers u and v representing an edge that connects node u and v.

If there is no such graphs, output `-1 -1` in a single line.

Sample Input 1	Sample Output 1
5 6 1 5 2 3 4	1 5 2 3 5 2 4 3 3 5 4 5

Explanation of Sample 1: You can also output the following edges and get a correct answer: $(1, 5), (5, 2), (2, 3), (3, 5), (2, 4), (5, 4)$.

Sample Input 2	Sample Output 2
5 10 1 5 2 3 4	-1 -1

Problem C
Polynomially Constructed

You are given an integer N and an array of **integers** $[y_1, y_2, \ldots, y_N]$ of size N.

You want to construct the following:

- a polynomial $P(x) = c_0x^0 + c_1x^1 + \ldots + c_dx^d$ where the coefficients are **real numbers**, and for some *degree* d;
- an array of **real numbers** $[x_1, x_2, \ldots, x_N]$ of size N such that $x_1 < x_2 < \ldots < x_N$;

and the construction satisfies $P(x_1) = y_1$, $P(x_2) = y_2$, …, $P(x_N) = y_N$.

It can be proven that such a construction always exists. Your task is to find the minimum polynomial degree d that you can construct.

Input

The first line contains an integer N ($1 \leq N \leq 100\,000$). The second line contains N integers representing y_i ($-10^9 \leq y_i \leq 10^9$).

Output

Output the minimum degree d in a single line.

Sample Input 1	Sample Output 1
5 2 -1 1 1 0	3

Explanation of Sample 1: We can construct a polynomial $\frac{29}{3}x^0 + 8x^1 + 0x^2 - \frac{2}{3}x^3$, and array $[-2.81273..., -2, -1.24361..., 3.86937..., 3.91423...]$.

Sample Input 2	Sample Output 2
5 1 0 0 0 1	4

Sample Input 3	Sample Output 3
3 0 0 0	0

This page is intentionally left blank.

Problem D
Divido Travel

There is a connected weighted graph with N vertices, numbered from 1 to N, and M edges, numbered from 1 to M. Edge i connects vertex U_i and vertex V_i bidirectionally with weight W_i.

You want to travel from vertex 1 to vertex 2, then from vertex 2 to vertex 3. You are allowed to visit any vertex or any edge more than once. The cost of your entire travel is the sum of the weights you passed through.

There is, however, a catch, in which every time you pass through an edge, its weight will be halved then rounded up to the nearest integer. Formally, if its weight is previously w, then it will change into $\lceil \frac{w}{2} \rceil$ after you pass through it.

Calculate the minimum cost of your entire travel.

Input

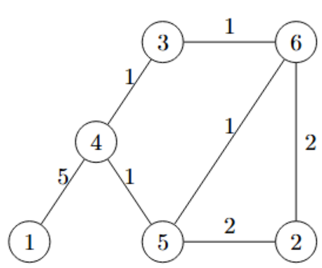
The first line contains an integer N and M ($3 \leq N \leq 100\ 000$; $N - 1 \leq M \leq 200\ 000$). Each of the next M lines contains U_i, V_i, and W_i ($1 \leq U_i < V_i \leq N$; $1 \leq W_i \leq 10^9$) describing an edge. The graph you are given contains no multi-edges.

Output

Output the cost of your entire travel in a single line.

Sample Input 1	Sample Output 1
6 7 5 6 1 4 5 1 3 4 1 1 4 5 2 5 2 3 6 1 2 6 2	11

Explanation of Sample 1: The graph can be illustrated as follows.



You can travel from vertex 1 to vertex 2 by traversing the vertices $1 \to 4 \to 5 \to 2$. The weights of the traversed edges are changed into 3, 1, and 1 respectively.

Then, you travel from vertex 2 to vertex 3 by traversing the vertices $2 \to 5 \to 6 \to 3$. The cost of the entire travel is $5 + 1 + 2 + 1 + 1 + 1 = 11$.

Sample Input 2	Sample Output 2
3 2 1 2 265 1 3 265	663

Problem E
Xordition Robot

You have a robot that contains N modules, numbered from 1 to N. Each module accepts an integer and outputs an integer. The output of module i becomes the input of module $i+1$ (for $1 \leq i \leq N-1$).

The specification of module i is either:

- `+ k`: given an integer x ($0 \leq x < 16$), the module outputs $(x+k) \bmod 16$; or
- `x k`: given an integer x ($0 \leq x < 16$), the module outputs $x \oplus k$, where $\oplus$ represents the bitwise XOR operator.

There are Q replacements, and the j-th is of the form:

- `i t k`: replace module i to a module with specification `t k`, where t is either `+` or `x`.

Each time a replacement is done, find the output of module N when module 1 is given an input 0.

Input

The first line contains two integers N and Q ($1 \leq N, Q \leq 200\ 000$). Each of the next N lines contains a character of either `+` or `x` followed by an integer k ($0 \leq k < 16$) representing the module.

The next Q lines contains an integer i ($1 \leq i \leq N$), followed by a character `+` or `x`, and finally an integer k ($0 \leq k < 16$), meaning that you have to replace module i to the specified module.

Output

Output Q lines, each containing the output of module N, after each replacement, when given an input 0 to module 1.

Sample Input 1	Sample Output 1
4 2 + 3 x 5 x 9 + 15 2 + 8 1 x 10	1 10

Explanation of Sample 1: After the first replacement, the modules are: `+ 3`, `+ 8`, `x 9`, `+ 15`

The output of module N is then $(((0+3)+8)\oplus 9)+15)$ is 1.

After the second replacement, the modules are: `x 10`, `+ 8`, `x 9`, `+ 15`

The output of module N is then $(((0\oplus 10)+8)\oplus 9)+15)$ is 10.

This page is intentionally left blank.

Problem F
Food Rating

In the new food delivery app *Touch*, customers are able to rate the driver with integer scores from L to R inclusive. Drivers will get a bonus based on how high their average rating is. However, some drivers may abuse this system. A driver delivering food to 1 customer may get 5.0 average rating, while another driver delivering food to 5 customers may get 4.8 average rating.

As the owner of the app, you need to ensure fairness in the bonus system. To do that, you need to know: for a driver to have an average rating of exactly X, what is the minimum number of delivery k, such that there exists a scenario where the average rating given by k customers is exactly X. In addition to that, output any list of k integers within L to R such that the average of the list is exactly X.

Input

The first line contains a real number X ($0 \leq X \leq 1000$). The number X contains at most 6 digits, including both digits before and after the decimal separator (if any).

The second line contains two integers L and R ($1 \leq L \leq R \leq 1000$).

Output

If there exists a scenario where a driver can get an average rating exactly X, output in the first line, the minimum integer k representing the minimum number of customers giving the rating. In the next line, output k integers between L and R representing the rating given by the customers.

If there is no such scenario, output -1 in a single line.

Sample Input 1	Sample Output 1
8.6 1 10	5 10 9 10 7 7

Explanation of Sample 1: The average of $[10, 9, 10, 7, 7]$ is exactly 8.6. It can be proven such that there is no valid list with four or less integers.

Sample Input 2	Sample Output 2
9 1 10	1 9

Sample Input 3	Sample Output 3
2.79 3 5	-1

This page is intentionally left blank.

Problem G
Mex XOR

You initially have an empty set S, and an integer K. You will then have to process Q queries, each giving you an integer X, meaning that you will have to **insert** X into S if $X \notin S$, or **remove** X from S if $X \in S$.

After each query, you would like to know the following. Find the minimum of MEX$(\{s \oplus i : s \in S\})$ for all $0 \leq i \leq K$.

The operator $\oplus$ is the bitwise XOR operation, while MEX is a function that returns the smallest non-negative integer that does not appear in the set. In particular, the MEX of an empty set is 0.

Input

The first line contains two integer Q and K ($1 \leq Q \leq 200\,000$; $0 \leq K < 2^{30}$).

Each of the next Q lines contains an integer X ($0 \leq X < 2^{30}$).

Output

Output Q lines, representing the minimum MEX value after each query.

Sample Input 1	Sample Output 1
4 2 1 0 2 1	0 0 1 0

Explanation of Sample 1: After the first query, the set S is $\{1\}$. We can see that MEX$(\{1 \oplus 0\}) = 0$, and this is the minimum possible value.

After the third query, the set S is $\{0, 1, 2\}$. The values to consider are as follows:

- MEX$(\{0 \oplus 0, 1 \oplus 0, 2 \oplus 0\})$ = MEX$(\{0, 1, 2\}) = 3$.
- MEX$(\{0 \oplus 1, 1 \oplus 1, 2 \oplus 1\})$ = MEX$(\{1, 0, 3\}) = 2$.
- MEX$(\{0 \oplus 2, 1 \oplus 2, 2 \oplus 2\})$ = MEX$(\{2, 3, 0\}) = 1$.

The minimum among them is 1.

After the fourth query, the set S is $\{0, 2\}$ and MEX$(\{0 \oplus 1, 2 \oplus 1\}) = 0$.

This page is intentionally left blank.

Problem H

Light Control System

There are N shops, numbered from 1 to N, that sell lamp switches. There are also M lamps, numbered from 1 to M, that are initially switched off.

When you buy and use a switch from shop i, it will **toggle all** lamps in the set S_i. When you toggle a lamp, it will become on if it is currently off, and it will become off if it is currently on.

You have to answer Q queries, each giving you a range L and R. You have to find the minimum $r - l + 1$, such that $L \leq l \leq r \leq R$ and there exists a subset $T \subseteq \{l, l + 1, \ldots, r\}$ and by buying switches from all shops $t \in T$, you can switch on all the lamps.

You have to answer each query or report that it is impossible.

Input

The first line contains three integers N, M, and Q ($1 \leq N \leq 50\,000$; $1 \leq Q \leq 100\,000$; $1 \leq M \leq 30$). Each of the next N lines contains an integer k_i ($1 \leq k_i \leq M$), followed by k_i integers between 1 and M representing S_i, containing the lamp numbers that can be toggled by buying a switch from shop i.

The next Q lines contain the queries, each giving you two integers L and R ($1 \leq L \leq R \leq N$) in a line.

Output

For each query, output an integer representing the minimum value, or -1 if it is impossible.

Sample Input 1	Sample Output 1
5 3 3 1 3 2 1 2 2 1 3 1 1 1 2 1 5 3 4 2 5	2 -1 3

Explanation of Sample 1: For the first query, you can pick $l = 1$ and $r = 2$, then you buy the switches from shop 1 and 2.

For the second query, there is no way to pick any l and r from the range $3 \leq l \leq r \leq 4$ such that you can buy switches that toggle all M lamps. Therefore you have to output -1.

For the third query, you can pick $l = 2$ and $r = 4$, then you buy the switches from shop 2, 3, and 4. It is also possible to pick $l = 3$ and $r = 5$, then you buy the switches from shop 3 and 5. Both ways give you $r - l + 1 = 3$.

This page is intentionally left blank.

Problem I
Rectangular Pool

You are the owner of a land that can be represented as a grid G of size $N \times M$, with rows numbered from 1 to N, and columns numbered from 1 to M. Define $G_{i,j}$ as the cell of the grid on the row i and column j. Each cell of G has either of the following terrains: puddle represented as `.` and dirt represented as `#`. A connected component of puddles are defined as a **pool**. A connected component is a set of cells such that any two cells in the set are connected by traversing between cells that share a side.

Your land is very bizarre that, for the next Q days, exactly one cell will change its type. Formally, suppose cell at row R and column C change its type. If the cell $G_{R,C}$ is a puddle, it will transform into dirt; if the cell is dirt, it will transform into a puddle.

You have an abnormal obsession to rectangles, so you will get sick if a pool is not rectangular. A rectangular pool is defined as follows: let r_{min}, r_{max}, c_{min}, and c_{max} as the minimum row number, maximum row number, minimum column number, and maximum column number of all puddles in the pool respectively, then there are exactly $(r_{max} - r_{min} + 1) \times (c_{max} - c_{min} + 1)$ puddles in the pool.

At the end of each day, determine if there exists any non-rectangular pool in your land!

Input

The first line contains two integers N and M ($1 \le N \le M \le 1000$). Each of the next N lines contains M characters of either `.` or `#`, where the character at i-th row and j-th column represents the type of cell $G_{i,j}$.

The next line contains an integer Q ($1 \le Q \le 300\,000$). Each of the next Q lines contains two integers R and C ($1 \le R \le N$; $1 \le C \le M$) meaning that the cell $G_{R,C}$ change its type.

Output

Output Q lines representing the existence of non-rectangle pool at the end of each day. Each of the lines contains either `RECTANGLES` if all pools are rectangular, or `NO` if there exists a non-rectangular pool.

Sample Input 1

```
5 5
#...#
#...#
#####
#...#
#####
3
4 3
1 3
2 3
```

Sample Output 1

```
RECTANGLES
NO
RECTANGLES
```

Explanation of Sample 1: The following are the states of the land after each day:

```
#...#   #.#.#   #.#.#
#...#   #...#   #.#.#
#####   #####   #####
#.#.#   #.#.#   #.#.#
#####   #####   #####
```

After the first day, all three pools are rectangular pools. After the second day, the pool formed by cells $(1, 2)$, $(1, 4)$, $(2, 2)$, $(2, 3)$, $(2, 4)$ is not rectangular. All pools are rectangular again after the third day.

Sample Input 2	Sample Output 2
3 4 #### #.## #### 1 2 2	RECTANGLES

Problem J
Mediation

You are the mayor of a tree-structured city with N districts, numbered from 1 to N, connected by $N-1$ roads, numbered from 1 to $N-1$. Road i connects district U_i and district V_i bidirectionally with weight W_i. Two districts S_1 and S_2 have been marked as **mediator districts**. The travel cost between district x and district y, denoted by $d(x, y)$, is the minimum sum of weights of the roads you need to pass through.

Whenever a conflict arise between any two districts, the mediator districts are required to travel to the conflicting districts. The **mediation cost** for two conflicting districts u and v, denoted by $M(u, v)$, is the maximum travel cost of the mediator districts to the nearest conflicting district. Formally, $M(x, y)$ can be calculated as follows.

$$M(u, v) = \max(\min(d(u, S_1), d(v, S_1)), \min(d(u, S_2), d(v, S_2)))$$

Calculate the sum of mediation cost $M(u, v)$ over all $1 \leq u < v \leq N$.

Input

The first line contains three integers: N, S_1, and S_2 ($2 \leq N \leq 200\,000$; $1 \leq S_1 < S_2 \leq N$).

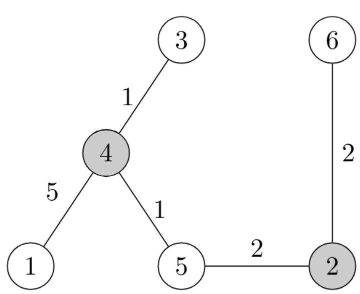
The next $N-1$ lines contains integers U_i, V_i, and W_i ($1 \leq U_i < V_i \leq N$; $1 \leq W_i \leq 100$) describing an edge.

Output

Output the sum of mediation cost in a single line.

Sample Input 1	Sample Output 1
6 2 4 1 4 5 3 4 1 4 5 1 2 5 2 2 6 2	35

Explanation of Sample 1: The city is illustrated as follows.



The values of $M(u, v)$ over all $1 \le u < v \le N$ are presented as follows.

u \ v	1	2	3	4	5	6
1		3	4	3	2	5
2			1	0	1	3
3				3	2	2
4					2	2
5						2

The sum of all $M(u, v)$ is 35.

Problem K
Palindrome Partition

A palindrome is a string that reads the same forwards as backwards. For example, `radar`, `noon`, and `a` are palindromes, while `bathtub`, `thought`, and `is` are not.

Given a string S consisting only of lowercase English letters. You can rearrange the letters in S in any order you like. Your task is to split S after rearrangement into as few palindromic substrings as possible.

Input

The first and only line contains the string S ($1 \leq |S| \leq 200\,000$) containing lowercase English letters.

Output

On the first line, print the minimum number k of palindrome substrings.

On the next k lines, print the palindromes that the string S (after rearrangement) can be split into. If there are multiple ways to split, you may output any of them.

Sample Input 1	Sample Output 1

`larcevalecer`	`2`
	`level`
	`racecar`

Explanation of Sample 1: We can rearrange the input string into `levelracecar`, then split the string into two palindromes: `level` and `racecar`. No rearrangement can produce a single palindrome, so the minimum number of palindromes is 2.

Sample Input 2	Sample Output 2
`abab`	`1` `baab`

Sample Input 3	Sample Output 3

`indonesianationalcontest`	`8`
	`i`
	`incni`
	`stats`
	`nnn`
	`ala`
	`odo`
	`t`
	`eoe`

This page is intentionally left blank.

Problem L
Honourable Arrays

You are given a set S containing N different positive integers. You are also given a positive integer K and a prime number M.

An array is said to be *honourable* if each of its elements is in S, and the product of all elements in the array is K modulo M.

For a given integer L, count the number of different honourable arrays with length L. Two arrays of length L are said to be different if there exists an index such that the elements in both arrays differ at that index. Output the count modulo $998\,244\,353$.

Input

The first line contains four integers N, K, M, and L ($1 \leq N, K < M$; $2 \leq M \leq 100\,000$; $1 \leq L \leq 10^9$; M is a prime). The second line contains N different integers representing S, each is a positive integer less than M.

Output

Output an integer representing the number of different honourable arrays modulo $998\,244\,353$.

Sample Input 1	Sample Output 1
2 1 3 4 1 2	8

Explanation of Sample 1: the different honourable arrays of size 4 are: $[1,1,1,1]$, $[1,1,2,2]$, $[1,2,1,2]$, $[1,2,2,1]$, $[2,1,1,2]$, $[2,1,2,1]$, $[2,2,1,1]$, $[2,2,2,2]$.

Sample Input 2	Sample Output 2
2 1 3 1000 1 2	510735315

This page is intentionally left blank.